Designing an Overlay Network for Modern Mobile Computers

Brandon Booth

University of Southern California
CS-555, Computer Science Department

April 29, 2011

1 Introduction

Trends in computer usage today reveal an industry shifting towards mobile devices. The number of mobile computing devices owned by people all over the world is growing rapidly too. Many of these devices, such as smart phones and tablets, are carried around by their owners throughout the day and are beginning to function as general purpose computers that provide a full range of services previously only accessible from personal computers in the office or at home. By re-examining applications designed for stationary personal computers, we can discover new ways to modify them to be more relevant in today's mobile world.

While many types of applications could benefit from this kind of change, peer-to-peer data sharing programs, such as Napster or FrostWire, are of particular interest because their performance can be improved by mobile technology. The network cost of providing data sharing services is reduced when they operate in a decentralized fashion, and when data can quickly be transferred. Distributed hash tables, especially those involving topologically-aware overlay networks, are great tools for providing the desired structured and decentralized file storage and retrieval. Existing topologically-aware overlay network designs, however, do not function adequately for a large number of mobile users who are spread out across the globe.

We contend that any well-designed overlay construction for mobile users includes these three qualities:

- Scalability - Should support a large number of users and also be capable of functioning world-wide
- Mobile Efficiency - Should minimize the amount of overhead incurred due to mobility and typical usage patterns for mobile users
- Good Network Performance - Should minimize the number of network hops required to store and access data and keep the request/response time as low as possible

Two current topologically-aware overlay constructions serve as a good basis for examining and evaluating existing solutions in

terms of these three qualities because each excels in different areas. These two systems are the Topologically-Aware Overlay Construction and Server Selection [1] and the Mobile Hash Table [2]. The next sections examine the suitability of this set of qualities and further describe why engineering an overlay network that satisfies all three is difficult. Each existing system is then evaluated with respect to these qualities in a mobile environment, and finally, further issues and other approaches to building a distributed hash table that better meets these goals are considered.

2 Important Qualities for Mobile Overlay Networks

In this section, we discuss why scalability, mobile efficiency, and network performance are preeminent considerations when designing an overlay network for a global-size, mobile overlay network. The difficulties involved with achieving all of these qualities in one system are also briefly explained.

2.1 Importance of Scalability, Mobile Efficiency, and Network Performance

Scalability is of fundamental importance for a number of reasons, mainly because our goal is to create a system that functions for all mobile users globally, but also because it positively influences other important qualities for overlay network designs. As the number of users increases in an overlay network, so can routing reliability and availability of files through the use of redundancy. When more nodes exist in the network, individual memory requirements for hosting file data are lessened and files distributed more equally, which makes it easier for the system to maintain file availability if a single node becomes over burdened or suddenly leaves the network.

Mobile efficiency is also essential because the network must be designed to handle users that change location without bogging down, halting, or in the worst case crashing other nodes due to upkeep costs. Mobile devices and their network communications are more fragile than desktop computers due to their reliance on battery power and wireless networking. A good mobile overlay network should account for this by reasonably minimizing the costs of joining, exiting, and traveling through the network.

Finally, good network performance comes from minimizing the number of hops required to transport a message because they dominate the cost of network communication. The motility of mobile devices suggests that a good overlay network needs to quickly ship information round-trip before a node's state changes. A fast communication channel reduces the difficult-to-manage effects of mobility and has the added benefit of enabling the system to quickly proliferate messages. Ensuring that the overlay network is very closely correlated to the underlying physical network topology is an important step towards achieving this performance goal.

2.2 Difficulties With Building A Global-Scale Mobile Hash Table

In general, scalability is difficult to achieve because it becomes harder to ensure that an overlay network remains functional as the number of users and the distances between them in-

creases. In a global-scale distributed system, individual nodes do not have the capacity to maintain an awareness of the state of the entire system. This means it is difficult to ensure that the system continues to function as a whole when lots of messages are being passed around over large distances. Since the risk of failure increases when many components are involved, achieving this goal requires careful error checking and handling, especially given the unpredictability of users in a mobile environment.

This unpredictability also makes it hard for an overlay network to keep overhead costs down to a minimum as users move around the network. Since network connectivity is constantly changing, keeping track of a node's neighbors in an ad hoc environment and which ones among them own particular pieces of data is challenging. As users move in and out of range of other users, existing data transfers may be re-routed or suspended, producing extra overhead for nearby mobile devices.

Re-routing data transfers also affects the performance of the overlay network in its attempt to minimize the total number of hops. In a globally-sized network containing millions of users, it is challenging to absolutely minimize the number of hops required between any two nodes because no one node knows the topology of the entire network. Using only a node's local knowledge, the best hope for routing messages efficiently is to make smart guesses and aim for a locally minimal hop count.

In short, the difficulty involved in building a globally-sized mobile hash table that achieves these three qualities is that they are each individually hard to achieve and also that they work against each other in part. In order to be successful, some degree of balance must be attained.

3 Evaluation of Existing Systems

This section evaluates existing overlay network constructions with respect to today's mobile world. In particular, it compares and contrasts each system's features against the ideal requirements for a good mobile overlay network: scalability, mobile efficiency, and good network performance.

3.1 Topologically-Aware Overlay Construction and Server Selection

The Topologically-Aware Overlay Construction and Server Selection (TOC) [1] system describes a method for matching a node's physical location in the world with its virtual location in the overlay network to improve network performance. The system as described is very scalable, but falls short on mobile efficiency and good network performance in a mobile environment.

The TOC is primarily based on the Content Addressable Network (CAN) [3], but achieves better performance through geographical awareness. The virtual overlay is still a $d-$dimensional torus, but further divides the space up into bins where each one loosely corresponds to a geographical area. Before joining the network, a node must measure its round trip times to a collection of servers with a priori known physical locations. Based on the relative results, it is assigned to a bin in the virtual space, and then joins the network in the same manner as the CAN by randomly picking its location inside the bin. The authors point out that this only leads to a rough geographical correlation between neighboring nodes in the overlay network, but mention that it still drastically improves performance on average [1].

Since the TOC is based primarily on the CAN, it is very scalable. In the CAN, each node is responsible for storing a fixed amount of data for managing the network structure, such as its location, region, and neighboring nodes, which means the overhead cost remains constant regardless of the number of users [3]. Furthermore, nodes joining the network split ownership of the surrounding region at the joined location by subdividing it, which enables new users in densely populated locations to immediately share a piece of the workload. Since nodes join at random locations within their corresponding bin, it is impossible to ensure that the region sizes are equal. As the number of users in each bin increases, the region sizes and thus the quantity of data storage requirements will probabilistically tend to equalize, so practically no nodes will be overly burdened or become central points of contention [3]. For small numbers of users, however, this becomes problematic, so the TOC should be used mainly at a larger scale.

The TOC was not designed with mobility in mind, so it is no surprise it exhibits unfavorable behavior in this case. This system is still interesting and useful to examine in a mobile world because it highlights problems that similar topologically-aware overlay networks might face. A bin in the TOC is the smallest resolution region representing a node's location in physical space. If a mobile node travels around in an area within the bin it is currently assigned to, then the TOC has no additional work to do to maintain the topological mapping. It is still as topologically-aware as was before since the node's position in the bin was random to begin with, and no discernible inefficiency is introduced because there were never any guarantees about the relative proximity of nodes within the same bin. When a mobile node crosses a bin boundary, there is no mechanism in the TOC design to explicitly deal with this case. If it aims to preserve the overlay's topological correlation with node location, then the node must migrate to a new bin. This migration can be achieved by having the node drop out of the network and then join again to be placed in the new bin. Preserving file availability in this case is a very costly operation because the node's current files have to be transferred to the neighbor that will reclaim the node's old region. When the node joins at the new location, it also has to split the surrounding region with another node and acquire the relevant files from it for storage. Both of these file transfer costs are magnified if multiple "realities" [3] are used too. If file availability is ignored, however, then the node pays no overhead penalty when leaving the network as the file copies can be avoided, but then that node's mobility causes files to suddenly disappear from the network - an undesirable side effect. The potential sum of overhead across the system for switching bins could otherwise be avoided by reducing the number of bins. This would increase the average communication cost between the randomly placed nodes in now larger bins, though, so this too should be avoided.

Additionally, tracking a mobile node would require that the round trip time to the known server locations be recomputed periodically. For large-scale systems, this would lead to network congestion and contention for this measurement and produce inaccurate results for new and existing mobile nodes, thus somewhat undermining the topologically-based bin partitions. Today's GPS-enabled mobile devices are able to accurately locate nodes even when there is much contention for the GPS service. With a little reworking, the TOC could use the GPS service to avoid this contention issue altogether.

Network performance in the TOC for a mobile world depends on a number of factors. In typical internet communication, each packet is

routed through some internet service provider, so the physical proximity of two nodes in the network does not necessarily correlate with increased performance [3]. In a mobile environment, this is not the case though as ad hoc networking can be used. Since the location in the TOC overlay network only roughly maps to physical space, there is no way to know whether nearby nodes are actually physically close. In this case, the lowest cost communication approach is to route packets through wireless access points (WAPs). In theory, if the overlay network contains a large number of small bins, and if we can be certain that each node in a given bin belongs there, then neighboring nodes would be able to communicate using ad hoc transmission and thus avoid the extra hops through the WAPs. It would be very difficult to achieve this level of precision though from a set of round trip ping time measurements since network congestion and other factors could affect the values used for placement. As long as the bin size is small enough that randomly placed nodes could use ad hoc communication, then the TOC could keep communication costs, WAP accesses, and congestion down considerably.

Overall, the TOC's overlay space remains very scalable, much like the CAN, but the overhead cost of maintaining its topological mapping is too high given the potential for inaccurate measurements and users moving across bin boundaries. The network performance is decent and can be improved further if the size of each bin decreases (and therefore the accuracy of the topological mapping increases), but then more overhead due to mobility is introduced to counteract the potential gain.

3.2 Mobile Hash Table

The Mobile Hash Table (MHT) [2] considers an efficient method for maintaining a topologically-aware overlay network for mobile devices. It designs an algorithm for deciding which mobile user is responsible for hosting data based on a users' physical locations and kinematic data. Mobile devices are assumed to be GPS-enabled and support ad hoc networking (MANETs), thus providing fairly accurate physical location coordinates relative to other mobile devices. Data for storage and retrieval is mapped to looping path trajectories where its position can be computed at any point in time. The mobile user most closely matching the data's trajectory is the one responsible for owning the data. The MHT is only partially scalable, but handles mobility efficiently while minimizing the number of network hops for mobile ad hoc networks.

The MHT runs on top of the Greedy Perimeter Stateless Routing (GPSR) [4] protocol and so is somewhat scalable in terms of quantity of users. GPSR is an ad hoc network routing protocol that is topologically-aware and uses a greedy forwarding algorithm to pass packets to a target destination at some physical location. Each node using GPSR is aware of nodes within its transmission range and frequently updates state information about each, including at least their physical locations. The amount of state data that any node has to store to track its neighbors is not strictly bounded. If mobile users are densely populated, the number of neighbor states a node must manage can easily be on the order of several hundreds and this only gets worse as the population density increases or as ad hoc transmission range technology improves. GPSR attempts to curb the overhead of updating this state cache by intentionally pruning neighbors where the communication link

crosses paths with other nodes' communication links [4]. This helps reduce the state size and associated overhead to a manageable amount, but one can still imagine a degenerate case where a node serves as the "hub" for many branches of the network graph. In a densely populated area, this type of scenario is extremely unlikely and if it occurred, it would only be a transient state given the highly mobile environment. Therefore, GPSR and the MHT can safely handle a large number of users, although at a greater performance cost due to the pruning operation.

Spatial scalability in the MHT is not guaranteed. Since the overlay network relies heavily on GPSR, it is limited to ad hoc network communication. Information sharing in this environment depends completely on a chain of pairwise reachable nodes between the requester and the destination. If no chain exists, then the destination is unreachable by the requester and the data is not available for storage or retrieval (but only temporarily until such a chain exists some time later). The effect of gaps in the network poses an issue for global spatial scalability as less densely populated regions such as deserts, oceans, or the country-side may contain too few or even zero ad hoc equipped devices. Ad hoc transmission range also plays a role in limiting the spatial scalability of MHT because it necessitates the existence of nodes spread out between a source and destination. The MHT is therefore suitable for metropolitan areas, but not for rural or sparsely populated areas.

Overhead caused by nodes joining and leaving the network is generally kept to a minimum in the MHT. Joining the network is efficient since the nodes use soft state to keep track of neighboring nodes. A new node only needs to broadcast a message others within its transmission range in order to announce its presence and request to join the network [2]. At that point, any files waiting for a suitable host node, or any files better accommodated by the new node are transferred over. Unfortunately, these file transfers need to happen immediately when the new node joins the network otherwise file availability is sacrificed. This is the price to pay for maintaining an efficient routing algorithm where the "best match" node is assumed to be the one responsible for managing particular data. Likewise, leaving the network sees a small amount of overhead cost. When a node simply drops from the overlay network, no action is necessary to repair the connectivity of it. In the more ideal case where a node announces its intent to leave the network, the only overhead includes transferring its data to other nodes before leaving. The lack of periodic "still alive" messages automatically removes the departed node from its neighbors' adjacency lists. If the network elects to use local redundancy to keep a node's data stored at some of its neighbors who also closely match the data's trajectory, then files need not be transferred from the node about to leave the network to preserve file availability. Using local redundancy also allows the synchronous file transfer of data to a node joining the network to be eliminated. So, the cost of joining and leaving the hash table network is small on average if redundancy is used.

The overhead due to mobility of existing nodes in the network is also kept to a minimum in most cases. The only maintenance overhead required is that of MHT's augmented GPSR protocol, which keeps an up-to-date list of nodes within transmission range along with their locations and trajectories. Rather than being updated on a frequent fixed interval, this list only needs to be updated periodically or when the trajectories change, which reduces the overhead cost. The representation of data as a looping trajectories in physical space enables the system to efficiently map data to a

node in motion. This results in data sticking with one mobile host for a longer period of time, even as the host moves around in physical space [2], and it is much more efficient than pairing data with a node's position alone. However, if the current host's trajectory deviates enough from the data trajectory, then the data is forced to migrate to a new host node that matches the data path better. This overhead is unavoidable if the system maintains that only one host be the primary server for a particular piece of data. This behavior is justifiable if node trajectories change less often than data requests are made as the system is correctly keeping the cost of locating the data to a minimum. For highly mobile nodes, though, using local redundancy would help alleviate the file transfer cost sometimes if the new host node already has the data, which may be more helpful overall.

As the MHT authors point out, the benefit of mapping data to node trajectories is only realized if the mobile users travel in predictable directions [2]. The MHT assumes that users move in straight lines, which is a very reasonable assumption in many cities and for users on the road. In a MANET, routing a response back to the requester is difficult because GPSR only routes to destination locations, not to mobile devices. Thus, nodes embed data about the requester's trajectory into the request packet so the reply makes it back to the correct location [2]. This solution is functional and efficient as little processing is required to recompute the return destination at each network hop, but it falls apart when the requester changes directions. When this occurs, the requester leaves a "buoy" file in the network that travels along its old trajectory and contains information about the new trajectory. When the reply packet is routed back to the requester's original projected location, the buoy is obtained from some node hosting the buoy data, and then the new trajectory is followed to reach the requester. This approach to managing changes in trajectories is very inefficient for a number of reasons: it requires suitable buoy hosts to exist in the network, it is unreliable since there is no guarantee that the buoy host will be available, and there is no limit to the number of buoys that can be left in the network if the requester changes directions frequently after sending the request. The existence of buoys can lead to greater network congestion as the density of nodes in an area increases since more buoys are likely to be stored. Buoy data is very important for ensuring that data is returned to the requester, so it needs to be prioritized for storage, lest the efforts of all nodes involved in fetching the data be undermined. This opens the network to denial-of-service attacks within a localized area if a node sends a request for data at a far away location and then changes directions a large number of times to flood the network with buoys. This malicious attack and the large amount of overhead it would produce for the system could be avoided by capping the number of times a node can change directions.

Packet routing in the MHT effectively minimizes the required number of hops. For nodes that are physically close, ad hoc networking performs better than wireless networking because zero extra hops are required to get to the destination, whereas wireless networking requires at least one hop to the wireless access point. For nodes that are somewhat farther away, GPSR's greedy forwarding approach is arguably the best choice for locally-aware ad hoc nodes, especially when the mobile user population is dense enough. However, in the case where the users are sparsely populated or where a "void" exists, forwarding becomes more inefficient as it requires routing around the void and eventually back towards the final destination [4]. Some additional network hops are introduced by GPSR's scalability constraint, which keeps it from storing

state data for every neighbor within transmission range as long as the network is not partitioned in the process. In some cases, this means the best neighbor node for forwarding a request using the greedy approach may not be reachable directly even though it is within transmission range, so at least an extra hop is introduced. This network performance inefficiency could be avoided if each node stored every neighbor's state, but this would affect the scalability of the design. Given that reduction in the number of neighbor states stored is necessary to ensure system scalability, this trade off is acceptable. An approach that strikes a better balance would be to prune neighbor nodes only if the total number of neighbors exceeds a manageable number. This would maximize routing performance while preserving scalability.

The network performance cost for frequent MHT operations is kept to an absolute minimum as well. When joining or leaving a network, a node only has to announce its intent to all neighbors within range and then possibly transfer files directly from or to those neighbors, incurring zero extra network hops. Changing directions during normal operation sees no additional hops because any more suitable node that would have the data transferred to it falls within transmission range. If no suitable node exists, then no data is sent until such a node appears, at which point the efficient greedy forwarding routing is again employed. Hot-spots, or data that becomes exceedingly popular, can cause its hosts to be congested thus increasing the response time for future requesters. Local redundancy aids the performance loss in this situation if a request passes through a neighboring node that also contains the file [2].

To summarize, even though MHT is not globally scalable and has some potential efficiency concerns for unevenly distributed populations, it manages mobility in a unique and appropriate manner. The mobile efficiency for users in only moderately populated regions is still very practical and the common case communication performance is extremely good for networks limited to ad hoc transmission.

4 Further Problems and Approaches to Better Mobile Overlay Construction

The TOC and the MHT both have some significant advantages and disadvantages when applied in a mobile environment. This section considers various approaches for improving the scalability, mobile efficiency, and network performance of mobile overlay networks.

First, we consider the use of ad hoc transmission and how it functions in a mobile world. Ad hoc network communication seems very appropriate for mobile networks mainly for its ability to route efficiently and reduce the number of hops. It performs very well when the number of users scales up since there are theoretically fewer voids in the network that need to be routed around. An increased user count also decreases the likelihood that the network becomes partitioned at any point in time. The reliability of the topologically-aware GPSR network improves for greater numbers of users because data is more likely to find a suitable node as its host.

So, ad hoc communication is a useful tool for mobile distributed hash tables, but it has some issues as well. With today's unevenly distributed world population, ad hoc communication alone cannot provide spatial scalability because the transmission range is limiting, so sending messages across oceans would be

impractical. The transmission range limitation can also cause inefficiencies due to people's movement patterns throughout the day. People tend to cluster in certain places at certain times, such as commercial districts during the day and residential districts at night. This leads to a significantly increased chance of voids in different areas at different times. Nodes in the sparse areas during later times of the day are responsible for hosting the files maintained by many more nodes during an earlier time. This inherently creates an unequal burden on different nodes and implies that the granularity of topological awareness might be too fine.

The potential to minimize network hops by using ad hoc transmission is likely only applicable over shorter distances. After some number of hops in a GPSR route, it would have been more efficient to use wireless communication instead. Supporting both types of communication would enable topologically-aware overlay networks to be spatially scalable and also improve average network performance by dynamically determining which type of routing to use. An appropriate heuristic for this choice would include a measure of distance to the destination and the expected population density during the message's trip (used to determine likelihood of a void). The biggest hurdle in obtaining this flexibility is that the wireless access points (WAPs) would need to be able to route to target destinations (like GPSR) rather than to IP addresses. Currently, we do not believe WAPs support this type of routing mode, or at least it is not for public use.

Additionally, we further examine how MHT's encoding of data as a position, velocity, and time tuple succeeds in handling node mobility where older position-only methods fail. This approach pays significantly fewer overhead message costs when nodes travel compared to the TOC because, by analogy, each node's region travels in a straight line along with it. This notion can be taken even further if an appropriate model for how people move is developed. The MHT's linear model is highly appropriate for certain environments (ie. grids, cities, highways), but some people can and still will switch directions often leaving a buoy in the network for each change and adding up to a large efficiency and performance hit overall. A predictive waypoint-based path encoding could save the overhead of dropping a buoy in the network as long as the sequence of waypoints remains correct. Since people have routines and tend to do the same kinds of things from day to day, a history of paths traveled (to and from work for example) could be used to build a probabilistically accurate waypoint path prediction. This approach would require a larger amount of space to be used in each message to store the waypoint-based path, but we believe this inefficiency would pale in comparison to the performance and reliability gains of leaving fewer buoys in the network.

The kinematic-based encoding of data has a downside, though, as it restricts the quantity of data that can feasibly be transported in one request. Efficient routing in the MHT overlay network requires that the return packets are able to find the requester node without having to locate many buoys. The larger the file request, the longer it takes to send the data back to the requester and the greater the chance that the requester's path will deviate from the original one causing buoys to be left behind. Large files should therefore be requested in chunks and could be stored in the network as a "header" file and smaller chunk files. To obtain the data, the user would locate the header file by requesting the large file by file name, then look up the chunk names from the returned header and request those separately.

Lastly, we consider the suitability of mobile device hardware for the goal of a globally-scaled distributed hash table. High-end multipurpose mobile computing devices today, such as the iPad2, have up to 64 gigabytes of storage space. With all types of freely-available applications available to users, eventually some of the many users will run out of space, and in turn affect the performance and reliability of the MHT. MHT assumes that the node with the best-fit trajectory for a piece of data is the one that stores and hosts it without regards to its memory limitations. A concrete system would need to implement some measures to deal with this. The system could drop the node from the network until memory is freed, but in the best case this would require a transfer of the data to neighboring nodes in order to preserve file availability. Perhaps a better approach would be to rely on local and global redundancy to provide storage for files that do not fit onto the low memory device, but still allow it to supply the overlay network with access to its files.

Coping with the shortcomings of both the MHT and TOC is a challenging task, but is necessary in order to construct an efficient and global-scale overlay network for the mobile world. An overlay network similar to the MHT that addresses the concerns expressed in this paper would be well-equipped to route efficiently and handle mobility well, but would still only be functional in denser population areas. The scalable TOC, with its hierarchical and easily sub-dividable regions, is highly complimentary to the MHT and would help it obtain global scalability.

A hybrid solution is one promising approach to constructing an overlay network that achieves all three goals. One could imagine a CAN-style overlay network functioning on an icosahedron with its regions defined by the triangular facets. The icosahedron would be mapped to the globe and nodes' locations determined by GPS. The regions would behave much like regions in the CAN, but when subdivided to a certain size, further subdivision halted and a MHT employed instead. Routing between regions would generally be long distance, so wireless communication would be used while routing within a MHT region would typically be over a shorter distance and thus ad hoc transmission used. This layout reasonably achieves both the mobile efficiency and minimal network hop goals while still providing a framework enabling a global-scale overlay network. File availability in this scenario for regions of little or no population (deserts, north or south poles, oceans, etc.) could be preserved by using multiple realities [3] and marking these regions as "dead". Each node would know a priori which regions in each reality are "dead" and simply not attempt to request data from them.

5 Conclusion

In this paper, we argue there are three primarily important qualities to consider when designing a mobile distributed hash table: scalability, mobile efficiency, and good network performance. While each of these are individually important, meeting all of these goals simultaneously is hard because to some extent they work against each other, so a balance needs to be achieved. Two existing systems are evaluated with respect to these qualities. The Topologically-Aware Overlay Construction and Server Selection [1] system is observed to provide spatial and user scalability, but lack the facility to manage mobility efficiently while keeping overhead and routing costs minimized. The Mobile Hash Table [2], on the other hand, adequately handles node mobility, locally minimizes network hops,

and even supports many users within a given space, but it has difficulty scaling spatially due to ad hoc transmission range limitations and varying population densities. A number of other problems and useful suggestions for improving these systems with respect to the three goals are discussed: the use of both wireless and ad hoc communication to improve network performance, historically-based and probabilistic path predictions to improve mobile efficiency, and local and global redundancy to combat issues that inevitably arise for large numbers of users. A hybrid approach for constructing an overlay network that combines both existing systems is briefly described offering a potential starting point for future research.

References

[1] Sylvia Ratnasamy, Mark Handley, Richard Karp, and Scott Shenker. "Topologically-Aware Overlay Construction and Server Selection". *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*. InfoComm, 2002.

[2] Olaf Landsiedel, Stefan Gotz, and Klaus Wehrle. "Towards Scalable Mobility in Distributed Hash Tables". *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, pages 203–209. IEEE Computer Society, 2006.

[3] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. "A Scalable Content-Addressable Network". SIGCOMM 2001, pages 161–172. ACM Press, 2001.

[4] Brad Karp, H.T. Kung. "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks". *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 243–254. MobiComm, 2000.

[5] Junghee Han, David Watson, and Farnam Jahanian. "Topology Aware Overlay Networks". *Twenty-Fourth Annual Joint Conference of the IEEE Computer and Communications Societies*. InfoComm, 2005.

[6] Miguel Castro, Peter Druschel, Y. Charlie Hu, and Antony Rowstron. "Topology Aware Overlay Networks". *Microsoft Technical Report*, 2002.

[7] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan. "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications". *IEEE/ACM Transactions on Networking*, pages 17–32. IEEE Press, 2003.